REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
16-11-2010	Technical Paper	NOV 2010 - DEC 2010

4. TITLE AND SUBTITLE

Characterizing Open Source Routing Radio-to-Router Information in an Airborne Network

5a. CONTRACT NUMBER
FA8720-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Bow-Nan Cheng, Randy Charland, Paul Christensen, Andrea Coyle, Edward Kuczynski, Stephen McGarry, Igor Pedan, Leonid Veytser, and James Wheeler

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

ESC/CAA
20 Schilling Circle, Bldg 1305
Hanscom AFB, MA 01731

10. SPONSOR/MONITOR'S ACRONYM(S)

ESC/CAA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Although emulation environments provide a baseline for how systems will perform in real life, field-tests are crucial to demonstrate capabilities in real-world operating environments. In this paper, we describe and characterize an open source router implementation with extensions to support OSPFv3 MANET designed router, OSPFv3 dual stack address families, OSPF link metrics cross layer support, simplified multicast forwarding (SMF), and a radio-to-router interface performance in an airborne environment. Three airborne assets participated in the exercise to form a high capacity aerial backbone made of heterogeneous radio technologies and operating parameters. The assets and radio technologies formed dynamically routed airbridge 250 Nm and allowed the passing of military operational data.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	SAR	10	Zach Sweet
U	U	U			19b. TELEPHONE NUMBER *(include area code)* 781-981-5997

Characterizing Open Source Routing Radio-to-Router Information in an Airborne Network

Bow-Nan Cheng, Randy Charland, Paul Christensen, Andrea Coyle, Edward Kuczynski,
Stephen McGarry, Igor Pedan, Leonid Veytser and James Wheeler
Airborne Networks Group
MIT Lincoln Laboratory
{bcheng, rcharland, paulc, coyle, edk, smcgarry, ipedan, veytser, jwheeler}@ll.mit.edu

***Abstract*—Although emulation environments provide a baseline for how systems will perform in real life, field-tests are crucial to demonstrate capabilities in real-world operating environments. In this paper, we describe and characterize an open source router implementation with extensions to support OSPFv3 MANET designed router, OSPFv3 dual stack address families, OSPF link metrics cross layer support, simplified multicast forwarding (SMF), and a radio-to-router interface performance in an airborne environment. Three airborne assets participated in the exercise to form a high capacity aerial backbone made of heterogeneous radio technologies and operating parameters. The assets and radio technologies formed dynamically routed airbridge 250 Nm and allowed the passing of military operational data.[1]**

I. Introduction

The presence of a stable, high capacity network infrastructure has become a necessity in recent years due to the growth of net-centric applications in the private and public sector. As a result, it has become increasingly important to extend this infrastructure in a dynamic way in the absence of a stable wireline network. Although there are several approaches to providing an on-demand high capacity network in the absence of network infrastructure, we focus on the using of a high capacity airborne network to supplant physical network infrastructure. Figure 1 illustrates the concept and the test: by extending the range of communications through a high capacity airborne network, connectivity is maintained end-to-end without the presence of a physical network. The airborne layer, however, is not without its issues.

The current generation of radio systems fielded both in industry and the DoD community follow a stove-piped, operator assisted, model with each radio designed for a specific task and interoperable with only a small subset of even similar systems. While functionality in a homogeneous network of identical systems works well, platforms often employ a heterogeneous range of communication systems, each with its own proprietary interface to command and control, require manual setup and configuration, and require significant resources to route over each system in a dynamic way. Airborne networks employing high capacity radio technologies are no exception.

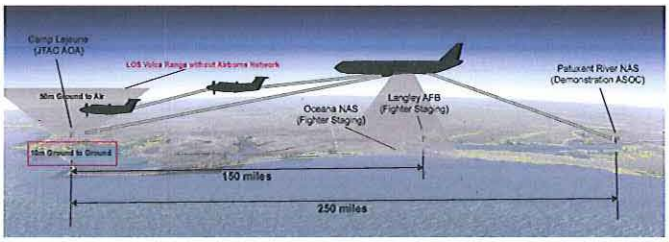
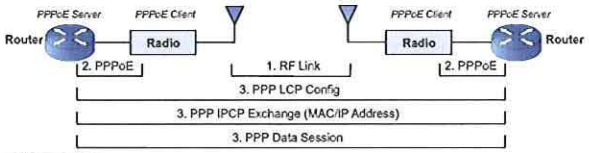


Fig. 1. Communications range extension through a high capacity airborne network



Connection Steps

1. **RF Link Establishment** – When a radio comes into contact with another radio, it forms a link. The process is radio dependent.
2. **PPPoE Radio-to-Router Establishment** – When a radio link is established, a PPPoE client on the radio initiates a PPPoE session with the corresponding router for both radios
3. **PPP Session Establishment** – Once the PPPoE Session is established, the PPPoE server on the router uses LCP packets to establish an end-to-end PPP connection. These packets are passed through the radio and over the RF link to the other PPPoE server on the other side of the radio.

An end to end PPP session is now created between 2 routers

- **Link Quality** – PADQ packets are periodically sent from each Radio to the Router, providing instantaneous link quality information
- **Flow Credit Report** – PADC/PADG packets are periodically sent from each Radio to Router to provide flow control
- **Virtual Multipoint Interface (VMI)** – Aggregates all PPP links into one virtual interface that provides broadcast/multicast emulation.

Fig. 2. Link setup via PPPoE RFC4938

To mitigate issues of manual static setup and lack of a common software interface to the heterogeneous radio systems, several vendors [1], [2] have proposed a standard radio-to-router interface - by extracting link information such as data rate, latency, and relative link quality from each radio and providing a common interface (with additional link informa-

[1]This work is sponsored by the United States Air Force under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, recommendations and conclusions are those of the authors and are not necessarily endorsed by the United States Government.

tion) to the router, routing protocols can make more intelligent routing decisions based on instantaneous layer 2 information. To the router, all it sees are a bunch of links with a specific subset of link information. RFC4938 describes this interface in detail and Figure 2 illustrates the basic concept. When an RF link is established, the radio, running a point-to-point protocol over ethernet (PPPoE) client initiates a PPPoE session with the router running a PPPoE server. When this is established, the router negotiates an end-to-end PPP connection with the node router on the other side of the link. At periodic intervals, the radio probes the link and sends to its router PADQ packets containing link information such as per link latency, current and maximum data rate, relative link quality, and neighbor up/down state. From this information, the routers dynamically calculate a link cost based on the quality of the link and the congestion, and weights the path choices accordingly. The radio periodically sends credits and credit grants to the server to throttle the rate of data sending from the router to the radio. As credits are used up, the packets are queued per link.

Although utilizing RFC4938 radio-to-router interface and building open source routers have been shown to be useful in the emulation environment and rapid prototyping, it is interesting to understand its performance in a real life airborne network. In this paper, we present results of a flight test involving three aerial platforms and two ground stations employing heterogeneous radio systems spread over 250 miles between Cherry Point, NC and Patuxent Naval Air Station, MD. Each of the air and ground platforms had a combination of three major high capacity radio systems and two commercial satellite systems.

The main test focused on characterizing the high capacity radio systems and testing the RFC4938 radio-to-router interface and the open source router. The open source router running on a Debian 4.0 Linux machine comprised of a Quagga [3] 0.99.9 base version modified to support OSPFv3 MDR [4], address families [5], dual stack IPv4 and IPv6 address families, and link metrics for cross layer information. Additionally, an open source common virtual multipoint interface [2] was implemented along with a simplified multicast forwarding (SMF) [6] for multicast traffic. The flight test spanned two weeks and we report the results from two of the six actual flight days (day1 and day 2).

In this paper, we present the measured base radio system characteristics as well as the interaction of the radio-to-router interface with the open source router. Section II describes the setup of the flight test including measurement techniques while Section III presents and briefly discusses the measured results. Finally, Section IV concludes the paper.

II. Experiment Test Methodology

To support the goal of creating a high capacity aerial layer to provide communications in challenging situations, the flight exercise brought together three air assets and two ground assets each housing a different combination of heterogeneous radio systems (see Figures 3 and 4) to bridge network connectivity

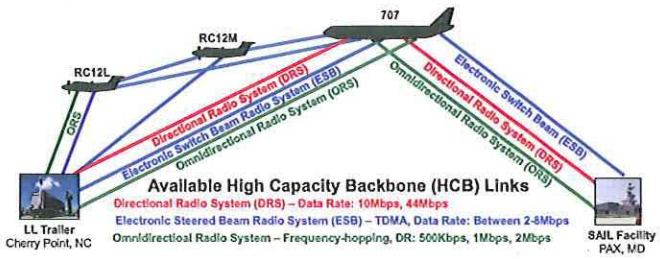


Fig. 3. Available High Capacity radio systems during flight exercise

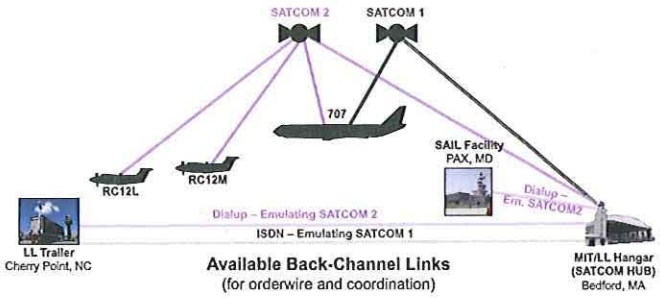


Fig. 4. Available SATCOM radio systems during flight exercise

over the span of 250nm. Each radio system and its general characteristics are listed below:

- **Directional Radio System (DRS)** - A highly directional point-to-point radio system with operating data rate of 10 and 44Mbps and an average of 3ms link latency
- **Electronic Switch Beam Radio System (ESB)** - A time division multiple access (TDMA) electronic switch-beam (ESB) radio system with operating data rate between 2 and 8Mbps and an average of 150ms link latency
- **Omnidirectional Radio System (ORS)** - A frequency hopping spread spectrum omnidirectional radio system with built-in layer 2 routing, an operating data rate between 500Kbps and 2Mbps and an average of 70ms link latency
- **Satellite System 1 (SATCOM 1)** - A commercial satellite system with an operating data rate of 128Kbps and a link latency of roughly 500ms
- **Satellite System 2 (SATCOM 2)** - A commercial satellite system with an operating data rate of 2400bps and a link latency of roughly 2 seconds

The main test focused on the high capacity radio systems while the SATCOM radios offered a back channel for test coordination and orderwire data. To characterize and measure the link and effectiveness of the radio-to-router interface and open source router, several data collection mechanisms were employed on multiple levels. Data collection was broken down into several suites physically located on different machines that either polled for data at set intervals or was event driven. Data was characterized as either useful for network characterization or for software debugging. Figure 5 describes

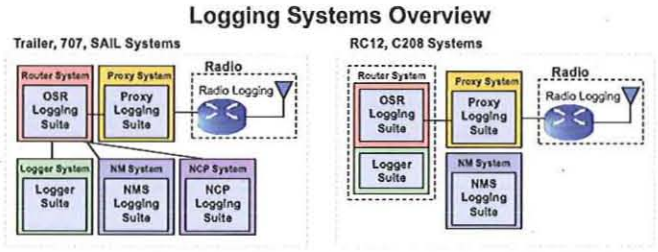


Fig. 5. Data collection suites on various platforms

the basic logging infrastructure. Due to importance and space limitations, we focus primarily on two major logging suites: the Proxy Logging Suite and the Open Source Router Logging Suite. The following list describes the basic functions of each component of the proxy logging suite:

- **RFC4938 Client Daemon Logs** (Event driven) - A record of rfc4938 daemon messages dumped from /var/log/syslog. The information contained in these logs helps to understand start/stop of PPPoE sessions and errors that might occur
- **Neighbor Monitor Logs** (Event driven) - Dump of neighbor monitor debug events. Information contained in these events help determine results from individual radio query threads as well as RF link status
- **Ping Priority Logs** (Event driven) - A record of the QoS prioritization of packets leaving the proxy system for the radio. We wanted to make sure that periodic link pings to determine neighbor up/down over legacy links always had priority

Similarly, the following list describes the basic functions of each component of the open source router logging suite:

- **IPv6 Link Local Pings** (1 second interval) - Time-stamped prioritized 16B pings sent from all Quagga routers to all Quagga routers through each link/cvmi IPv6 link local interface. The purpose is to establish per link uptime and measured link latency
- **Zebra Routing Table** (1 second interval) - Per second dump of Zebra routing tables to determine Zebra routes and correlate with Kernel routes
- **Zebra Debug Log** (Event driven) - Dump of Zebra debug logs for instantaneous, event driven information for statics generation and Zebra code debug
- **OSPF Neighbor Info** (1 second interval) - Per second dump of OSPF neighbor state table, LSDB, neighbor metric/cost, MDR level, link metrics reported per interface. The information is used to understand OSPF metric relation to physical link up/down
- **OSPF Debug Log** (Event driven) - Dump of OSPF debug logs for instantaneous, event driven information for statistics generation and OSPF code debug
- **PADQ Metrics** (Event driven) - A record of PADQ metrics per link as reported by CVMI to see what metrics are flowing to CVMI for each IPv4 and IPv6 neighbor. Data can also be used to correlate between link metrics seen by OSPF
- **PPPoE-Server Debug Logs** (Event driven) - A record of what PPPoE-Server is doing via /var/log/syslog. The information is used to debug PPPoE-Server issues
- **PPP Debug Logs** (Event driven) - A record of event driven PPP logs used to debug PPP issues
- **Kernel Routing Table** (1 second interval) - Per second dump of the Linux IPv4 and IPv6 kernel routing table for use in determining routes seen by the Linux kernel
- **ESB Radio Log** (1 second interval) - Per second polling of Electronic Switch Beam Radio System (ESB) statistics (only offered by ESB). These statistics help in determining aircraft position, physical RF up/down state and additional information on link modulation and data rate
- **QoS Queues Dump** (1 second interval) - A per second polling of QoS queues to help determine how much of a certain class of data is being pumped through a network and what percentage is being dropped.
- **PPPoE Session and Discovery TCPDump** (Event driven) - A tcpdump of the PPPoE packets on the radio-to-router Ethernet interface. This information is used to help debug RFC4938 credit and PADQ issues
- **TCPDump of OSPF Control Packets** (Event driven) - A tcpdump of OSPF control packets (hello and LSA) on each CVMI interface. This information is used to compare control signaling overhead with and without OSPF link metrics without having to sort through GBs of packet dumps

III. Test Results and Discussion

In this section, we present the performance results of the radio-to-router interface and the open source routing solution. We evaluate metrics of network reachability, link and path latency, reported vs. measured latency per link, link up/down percentage and routing protocol overhead, under varying conditions of OSPF hello/dead interval, flight orbits and times, and several heterogeneous radio technologies.

A. Link Availability Analysis

In order to fully characterize how the open source router functioned, it was important to first understand the availability of the actual RF link. Link outages can be caused by a number of issues include airplane body blockage, bad antenna sectors, incorrect pointing, and RF/hardware failures. In this section, we present the basic link availability provided by the hardware links to establish a baseline to evaluate the radio-to-router interface and the open source router. The following subsections describe the availability of links, describing first how the radio-to-router interface performed in terms of PPP setup shortly after a physical RF link is up, followed by transition times between RF up and down vs. network up/down.

The 7 Layer OSI Model describes 7 layers of the network stack: 1) The physical layer which encompasses actual RF and signal coding techniques; 2) The link layer which describes the characteristics of a 1-hop link; 3) The network layer that

uses these 1-hop links to determine best end-to-end path; and layers 4-7 which describe more application-focused activities. Gathering statistics on each link builds on top of one another to form a coherent story.

By understanding the average RF uptime (layer 1), followed by the average PPP uptime (layer 2), and finally the average OSPF neighbor uptime (layer 3), as well as the time it took to transition from one state to another, we can fully understand how effectively our developed software operated. For ESB, the RF link up and down were reported directly by the radio. For the other radios, we measured link "up" and "down" with 1 second link pings from proxy to proxy. Although not an exact representation of RF up, proxy-to-proxy link pings give us an easy way to determine radio link up given very little insight into the radio itself. PPP up was determined by IPv6 link local prioritized pings across each PPP link and OSPF neighbors were determined by querying the OSPF6d process. All the data was polled in 1-second intervals with OSPF hello and dead timers set to 10 and 40 respectively.

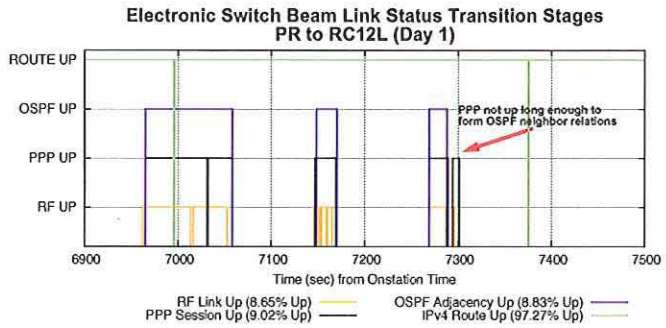


Fig. 6. Radio System 2 RF, PPP, and osPF link status from PR to RC12L (air to air) over 10 minute interval. There are times when a PPP is not up long enough for OSPF neighbor relations to form

Figure 6 give representative descriptions of our observations with the radio-to-router interface for the Electronic Switch Beam Radio System (ESB) during a 10 minute cross-section. In short, although there are fluctuations in the actual RF uptime, for the most part, a PPP is maintained to ride out small RF outages. When an end-to-end PPP is established, OSPF neighbor relations occur within 2 seconds. The 2 second delay is because when a PPP link is established, the CVMI sends a "link up" message to the Quagga router which then instructs OSPF to initiate an immediate "hello" and reset hello timers. Since layer 2 has no information about router-IDs described for layer 3 routing decisions, this OSPF hello exchange is necessary. As can be seen in Figure 6, for a short period around 7280 seconds, even though the RF and PPP link is up, it is terminated too quickly for OSPF neighbor relations to form.

Tables I and II show the average RF, PPP, and OSPF neighbor uptime on two flight days. For the most part, the proxy performed well in that when an RF link was up, a PPP was immediately established. The largest portion of difference between RF up and PPP up with the DRS and ESB radios was mainly due to a software issue that caused a PPPoE process to hang as can be seen with the high discrepancy between RF availability and PPP availability between PR and TR using ORS on both days.

TABLE I
AVG. RF, PPP, AND OSPF NEIGHBOR UPTIME - DAY 1

Platform (Radio)	Avg RF Up	Avg PPP Up	Avg OSPF Nb
PR - SAIL (DRS)	95.0%	84.1%	82.8%
PR - TR (DRS)	88.0%	83.2%	82.6%
PR - SAIL (ESB)	–	–	–
PR - RC12L (ESB)	9.50%	9.84%	9.57%
PR - RC12M (ESB)	31.3%	31.8%	31.6%
RC12L - TR (ESB)	76.1%	78.7%	78.6%
RC12L - RC12M (ESB)	8.40%	8.35%	7.39%
PR - SAIL (ORS)	99.8%	99.4%	99.4%
PR - TR (ORS)	99.6%	76.7%	76.4%
RC12L - TR (ORS)	–	–	–

TABLE II
AVG. RF, PPP, AND OSPF NEIGHBOR UPTIME - DAY 2

Platform (Radio)	Avg RF Up	Avg PPP Up	Avg OSPF Nb
PR - SAIL (DRS)	99.3%	100%	100%
PR - TR (DRS)	90.7%	88.5%	87.8%
PR - SAIL (ESB)	81.3%	88.9%	86.8%
PR - RC12L (ESB)	34.7%	33.1%	30.0%
PR - RC12M (ESB)	52.8%	53.0%	50.9%
RC12L - TR (ESB)	72.4%	74.4%	68.7%
RC12L - RC12M (ESB)	6.88%	7.01%	9.17%
PR - SAIL (ORS)	–	100%	100%
PR - TR (ORS)	88.0%	35.0%	31.3%
RC12L - TR (ORS)	41.0%	35.2%	29.6%

An interesting observation is that for some cases, notably for ESB, average PPP uptime is higher than the average RF uptime. This is due to the ESB having built-in, configurable time-out before a PPP link is removed once the radio transitions from a "good" to a "poor" RF state. In our tests, this was set by the vendor to 10 seconds and done in order to ride-out potential small changes in RF link quality. In short, if setting up a PPP took 2 seconds and the RF link outage was only 1 second, there would be no need to tear down the PPP link as it would take twice as long to re-establish the PPP connection.

For our proxy implementation, a configurable PPP initiate and tear-down time was set to 3 seconds in order not to thrash the rfc4938 client process. In short, whenever the neighbor monitor determined a link to be up, it waited 3 seconds before initiating a PPPoE session with the router. Similarly, whenever a link was down, it waited 3 seconds before terminating the PPPoE session. This leads to lower PPP uptime vs. RF uptime as shown in the data tables. Additionally, there were times when the PPPoE process on the proxy system would hang resulting in lower PPP uptime.

B. Link RF to PPP to OSPF Neighbor Transition Analysis

In addition to understanding RF, PPP, and OSPF neighbor uptime, we were interested in characterizing how quickly a PPP link was established over various radios when an RF link was detected to be "up". Table III and Table IV document our results over both flight days. It is important to note that

average RF up to PPP up was calculated as the time the difference between the RF and PPP link coming up, taking into consideration the 3 second hysteresis our proxy system employed. This only applied to all radios except ESB as the ESB had RFC4938 built in natively to the radio. In our analysis, outliers where it took 20 seconds or more for a PPP to come up after an RF came up were ignored. The average PPP up to OSPF neighbor and link ping up represents the average time after a PPP session is established for OSPF neighbor relations to be formed and link pings to be able to go through the network. On average, its expected that link pings should be quicker than OSPF neighbor establishment. The average of the time was taken over all the flight days over all platforms and the 95% confidence intervals given.

TABLE III
AVERAGE RF TO PPP TO OSPF UP TRANSITION TIME

Radio System (Avg of 2 days)	Avg RF Up to PPP Up	Avg PPP Up to OSPF Nb Up	Avg PPP Up to Link Ping Up
DRS	0.81 ± 0.299s	0.421 ± 0.263s	0.265 ± 0.127s
ESB	0.54 ± 0.039s	0.319 ± 0.035s	0.349 ± 0.029s
ORS	1.96 ± 0.268s	0.758 ± 0.224s	0.816 ± 0.227s
SATCOM 1	2.50 ± 0.523s	0.689 ± 0.289s	0.806 ± 0.258s
SATCOM 2	2.07 ± 0.368s	0.742 ± 0.227s	–

TABLE IV
AVERAGE RF TO PPP TO OSPF DOWN TRANSITION TIME

Radio System (Avg of 2 days)	Avg RF Dn to PPP Dn	Avg PPP Dn to OSPF Nb Dn	Avg PPP Dn to Link Ping Dn
DRS	0.058 ± 0.032s	0.108 ± 0.110s	0.000 ± 0.000s
ESB	2.701 ± 0.102s	0.254 ± 0.046s	0.002 ± 0.002s
ORS	0.205 ± 0.098s	0.039 ± 0.017s	0.013 ± 0.009s
SATCOM 1	0.208 ± 0.128s	0.728 ± 0.682s	0.000 ± 0.000s
SATCOM 2	0.228 ± 0.084s	0.189 ± 0.141s	–

As shown in Tables III and IV, for the most part, average transition times were very short (less than 1 second) with the exception of the SATCOM links. This is due to the high latency over the links.

C. Link Latency / Data Rate and Affect on OSPF Cost Analysis

In this flight test, the dynamic OSPF cost generated for each path is a function of the radio's reported current data rate (CDR) and latency. If the latency or CDR fluctuates often, the resulting OSPF cost assigned to the link will vary as well causing link flapping. It therefore becomes important to understand how each radio behaves in terms of CDR and latency. In the following subsections, we characterize the metrics used to calculate the OSPF cost for each radio and compare the calculated cost over various weights.

1) PADQ Link Latency: The RFC4938 specification defines a PADQ packet that allows radios to report latency to the router. Radios such as the ESB have built-in mechanisms to calculate latency while non-RFC4938-compliant radios measure the latency from proxy to proxy via link pings and report this information back to the router in the PADQ. It is interesting to see the variability between reported latency vs. measured latency on the ensuing PPP link formed end-to-end. We measure the latency using IPv6 link local pings to ensure they go over the same path and half the resulting return time. Figures 7-9 to compares the measured latency vs. the PADQ reported latency and the distance of the assets. We attempt to show representative results.

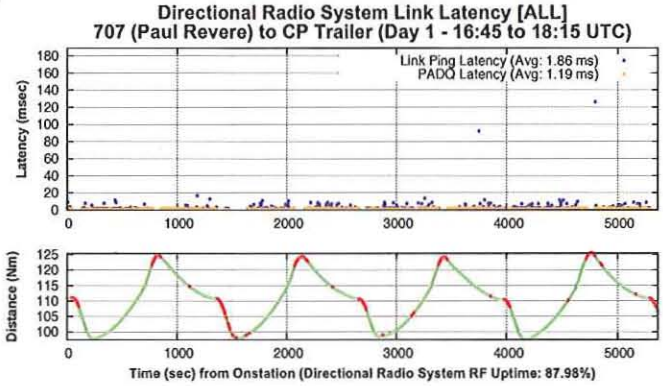


Fig. 7. Representative reported PADQ latency vs. measured latency for Directional Radio System. For the most part, reported latency was a little lower than measured latency.

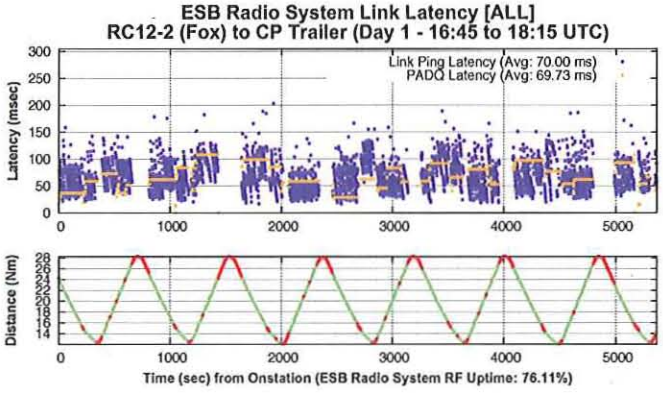


Fig. 8. Representative reported PADQ latency vs. measured latency for Electronic Switch Beam Radio System. For the most part, reported latency was a little lower than measured latency.

As seen in Figures 7-9, all the radio systems' latencies were fairly stable across the board with the PADQ reported latency slightly smaller than the ping latency as expected. ESB average latency is much higher than the DRS latencies due to its time division multiple access (TDMA) scheme whereby frequency usage is sliced into timeslots. ORS showed higher latencies at certain points due to increasing load adding to collisions in the broadcast medium. Tables V and VI summarize the average ping time for each radio over time on both flight days.

2) PADQ Current Data Rate (CDR): Another factor that goes into the calculation of OSPF cost is the current data rate (CDR) as reported by the PADQ packet. Like with the latency, CDR is given by the ESB radio directly while the proxy queries the non-rfc4938 compliant radios for the information on a per-second basis. Figures 10-12 illustrate a few of the

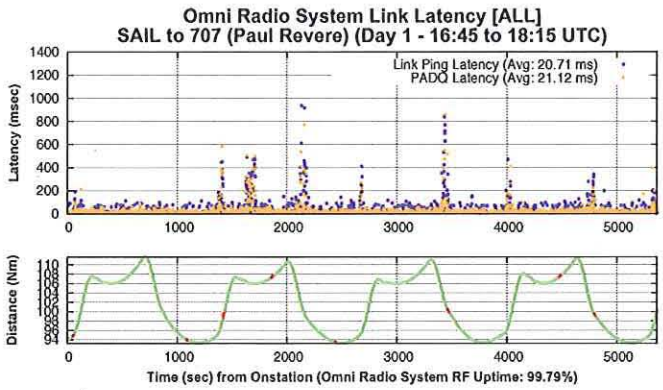


Fig. 9. Representative reported PADQ latency vs. measured latency for Omnidirectional Radio System. For the most part, reported latency was a little lower than measured latency.

TABLE V
COMP. OF PADQ REPORTED LATENCY VS. MEASURED LATENCY - DAY 1

Radio	Avg PADQ Latency	Avg Ping RTT/2
DRS		
- SAIL - PR	1.51 ± 0.02ms	1.98 ± 0.20ms
- PR - TR	1.19 ± 0.01ms	1.85 ± 0.18ms
ESB		
- SAIL - PR	–	–
- PR - RC12M	71.39 ± 0.62ms	69.75 ± 4.12ms
- PR - RC12L	63.27 ± 0.87ms	74.76 ± 7.02ms
- RC12M - RC12L	38.19 ± 2.28ms	63.83 ± 13.42ms
- RC12M - TR	78.80 ± 1.25ms	68.61 ± 15.50ms
- RC12L - TR	69.75 ± 0.67ms	70.00 ± 4.29ms
ORS		
- SAIL - PR	21.13 ± 1.50ms	20.71 ± 3.20ms
- PR - TR	34.90 ± 3.05ms	33.20 ± 6.56ms

data rates reported by the PADQ over the course of a day over various links in an air-to-ground configuration.

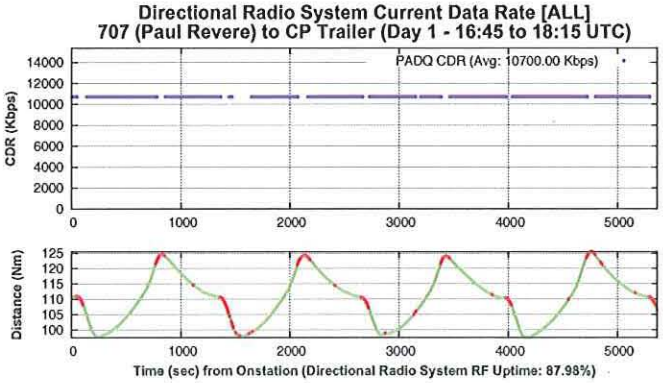


Fig. 10. Representative reported PADQ latency vs. measured latency for DRS. For the most part, reported latency was a little lower than measured latency.

As seen in Figures 10 to 12, DRS operated at 10.71 Mbps for the duration of the on-station time despite a few outages during the turns while ESB varied greatly in CDR. This is due to ESB's TDMA scheduling technique where it attempts to optimize frequency usage depending on the load, distance, and number of nodes in a network. The highly variable CDR

TABLE VI
COMP. OF PADQ REPORTED LATENCY VS. MEASURED LATENCY - DAY 2

Radio	Avg PADQ Latency	Avg Ping RTT/2
DRS		
- SAIL - PR	1.34 ± 0.04ms	1.80 ± 0.17ms
- PR - TR	1.10 ± 0.06ms	1.75 ± 0.15ms
ESB		
- SAIL - PR	51.34 ± 1.37ms	84.68 ± 10.10ms
- PR - RC12M	64.15 ± 1.37ms	61.81 ± 7.75ms
- PR - RC12L	67.35 ± 1.30ms	59.97 ± 13.48ms
- RC12M - RC12L	43.44 ± 5.63ms	71.95 ± 25.96ms
- RC12M - TR	41.27 ± 4.15ms	65.92 ± 37.05ms
- RC12L - TR	60.93 ± 1.13ms	67.46 ± 7.00ms
ORS		
- SAIL - PR	17.29 ± 0.93ms	19.03 ± 3.37ms
- PR - TR	84.72 ± 10.17ms	45.45 ± 25.53ms
- RC12L - PR	108.64 ± 15.24ms	92.67 ± 53.29ms
- RC12L - TR	70.55 ± 8.44ms	77.67 ± 37.28ms

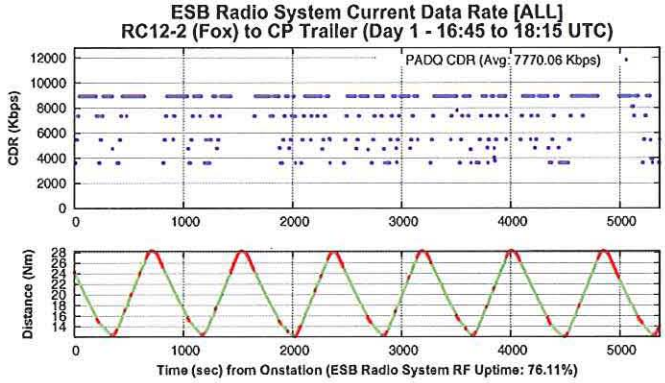


Fig. 11. Representative reported PADQ latency vs. measured latency for ESB. For the most part, reported latency was a little lower than measured latency.

suggests that OSPF cost for the ESB should vary quite a bit over time. It is also interesting to note that despite the link loss during the turns, DRS maintained a 87.98% uptime between the PR and TR while ESB maintained a 76.11% uptime between the RC12L and the Trailer. Due to the omnidirectional nature of ORS, it maintained 99.79% link availability between SAIL and PR.

TABLE VII
COMPARISON OF PADQ REPORTED CDR - DAY 1

Radio	Avg Data Rate	Std Dev Data Rate	Max Range
DRS			
- SAIL - PR	10.7 Mbps	0.0 Kbps	111.67 Nm
- PR - TR	10.7 Mbps	0.0 Kbps	124.34 Nm
ESB			
- SAIL - PR	3.61 Mbps	0.0 Kbps	107.16 Nm
- PR - RC12M	8.64 Mbps	877 Kbps	77.28 Nm
- PR - RC12L	4.80 Mbps	1490 Kbps	140.35 Nm
- RC12M - RC12L	6.77 Mbps	1433 Kbps	92.79 Nm
- RC12M - TR	5.12 Mbps	1869 Kbps	71.09 Nm
- RC12L - TR	7.77 Mbps	1800 Kbps	28.00 Nm
ORS			
- SAIL - PR	1.96 Mbps	171 Kbps	111.67 Nm
- PR - TR	1.61 Mbps	543 Kbps	127.03 Nm

3) Calculated OSPF Cost: The OSPF cost associated per link is a function of the reported PADQ latency and CDR from

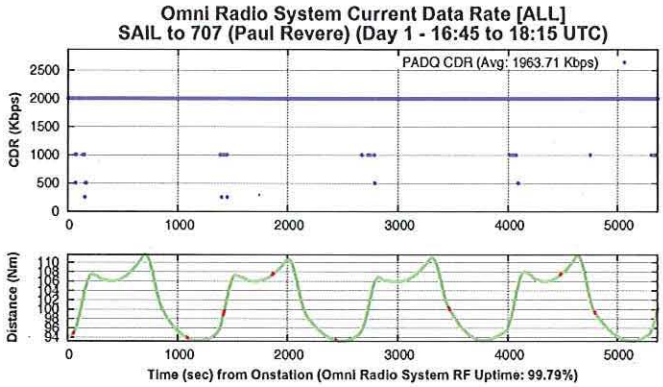


Fig. 12. Representative reported PADQ latency vs. measured latency for ORS. For the most part, reported latency was a little lower than measured latency.

TABLE VIII
COMPARISON OF PADQ REPORTED CDR - DAY 2

Radio	Avg Data Rate	Std Dev Data Rate	Max Range
DRS			
- SAIL - PR	10.7 Mbps	0.0 Kbps	111.15 Nm
- PR - TR	10.7 Mbps	0.0 Kbps	112.78 Nm
ESB			
- SAIL - PR	7.97 Mbps	1553 Kbps	96.09 Nm
- PR - RC12M	8.34 Mbps	1213 Kbps	51.87 Nm
- PR - RC12L	5.21 Mbps	1731 Kbps	73.19 Nm
- RC12M - RC12L	6.62 Mbps	1556 Kbps	69.64 Nm
- RC12M - TR	3.65 Mbps	267 Kbps	60.93 Nm
- RC12L - TR	7.57 Mbps	1987 Kbps	55.44 Nm
ORS			
- SAIL - PR	1.99 Mbps	170 Kbps	111.15 Nm
- PR - TR	1.22 Mbps	765 Kbps	112.78 Nm
- RC12L - PR	0.697 Mbps	448 Kbps	133.53 Nm
- RC12L - TR	1.26 Mbps	748 Kbps	53.26 Nm

either the radio or the proxy. These values are weighted on a scale of 0 to 100 with the sum of the latency weight (δ) and CDR weight (ϵ) adding up to 100. Equations 1 and 2 illustrate how OSPF cost is calculated and Figure 13 illustrates the basic concept: given the range of the radio latencies and operating data rates, a balance of latency and CDR was needed such that a small change in relevant latencies and data rates lead to big changes in cost while large changes in irrelevant latencies and data rates lead to small changes in cost. We leave a detailed analysis of the OSPF equation to other work and focus only on its usage.

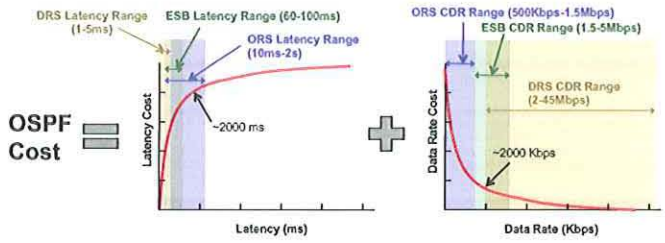


Fig. 13. OSPF Cost is a function of the link latency and current data rate (CDR)

$$OC = \left(\frac{\delta}{100}\right)\left(1000 \times (1 - e^{-0.0015 \times latency})\right) + \left(\frac{\epsilon}{100}\right)\left(1000 \times e^{-0.0015 \times CDR}\right) \quad (1)$$

$$\delta + \epsilon = 100 \quad (2)$$

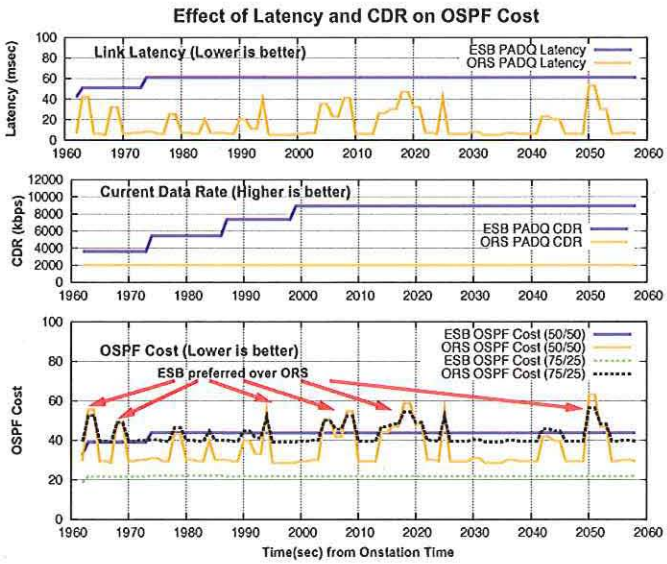


Fig. 14. Representative reported PADQ latency vs. measured latency for Radio System 3. For the most part, reported latency was a little lower than measured latency.

Figure 14 illustrates the effect of the reported PADQ latency and CDR on the OSPF cost based on a short snapshot of 100 seconds during the Day 2 flight between the RC12L and the Trailer. As shown, ESB has a much higher data rate than ORS, yet the ORS link has lower latency than the ESB. With the weights of CDR and latency set to 50 and 50 respectively, there are many times when the ESB link is preferred over the ORS link. This results in route flapping on otherwise 2 stable links.

When the CDR is weighted higher than the latency however (75/25), we notice that the dynamism in the ORS latency does not affect the OSPF cost enough to flap the links. The routing algorithm will always choose ESB if it is up because it has higher bandwidth. This issue was discovered on an earlier flight day when the RC12L and the Trailer ESB and ORS links were seemingly flopping back and fourth. The ORS dynamism in the latency seems to be due to ORS internal layer 2 routing decisions as other non-HCB ORS assets were in play as well as simply jitter on the latency. The immediate solution was the weight the CDR and latency in the OSPF cost formula to favor the higher bandwidth link. Additionally, we are taking a look at smoothing the proxy's reported latency and CDR over time in hopes of reducing the OSPF cost flapping.

D. Hello/Dead Interval Link Metrics Update Comparison Analysis

Any protocol modification design is a marriage of tradeoffs. In this section, we examine the tradeoff between having a low OSPF hello/dead timer vs. turning the link metrics feature on and off which allows the link layer to expose link up/down state and provide link metrics. We compare the overhead in packets and bytes vs. the network reachability. We define network reachability in two ways: 1) having kernel routes from all nodes to all nodes and 2) ability to ping from all nodes to all nodes. Having a route in the routing table does not necessarily guarantee that application data can flow through it as routes often need to converge. Our data below gives the kernel route network reachability and the ping network reachability for comparison.

During the flight test, these tests were performed once on Day 2. The test consisted of 4 distinct sub-tests each lasting 15 minutes at a different configuration (hello/dead interval at 1/4 and 10/40 and with link metrics on and off). No PPP's were forced to re-form and only OSPF configurations changed and the OSPF6d process restarted. When link metrics were turned off, the static OSPF costs assigned to each link were as follows: DRS - 1, ESB - 10, ORS - 20, SATCOM1 - 100, SATCOM2 - 1000. The tests were run with a script automatically switching configuration files at the exact second on all platforms.

OSPF Hello/Dead Interval Tradeoff

	Link Metrics ON	Link Metrics OFF
Hello Interval: 1 **Dead Interval: 4**	•High amount of hello packets •High amount of link state updates •Quick reaction to link changes •Dynamic link quality info	•High amount of hello packets •Low amount of link state updates •Quick reaction to link changes •NO Dynamic link info
Hello Interval: 10 **Dead Interval: 40**	•Low amount of hello packets •High amount of link state updates •Quick reaction to link changes •Dynamic link quality info	•Low amount of hello packets •Low amount of link state updates •Slow reaction to link changes •NO Dynamic link info

Fig. 15. The tradeoffs involved in tuning OSPF hello/dead intervals and link metrics

For each of our tests, the total bytes of sent control overhead (OSPF Hello, LSA, and DD packets) out each CVMI interface was compared with routes from all nodes to all nodes (reachability). Figure 15 describes the tradeoffs between each of the options. When OSPF Hello and Dead timers are low (1/4), a high amount of control overhead is flooded through the network to maintain links and routes. As a result, relatively quick reaction to link outages and routing changes take place at the cost of high overhead. When Hello and Dead timers are high (10/40), reaction to link outages happen slower (hello probes for neighbors "up" state happen only once every 10 seconds), but control overhead is relatively lower. Concurrently, when link metrics are turned on, dynamic link quality info is sent to the router to choose better paths. The result, however, is higher link state updates due to higher dynamic generated OSPF costs. When link metrics are off, no dynamic link information is shared with the router and costs are fixed.

Table IX describes the results of the test performed on Day 2. As expected, reducing the Hello and Dead interval to 1/4 resulted in a significantly higher number of control packets flooded network-wide. This was due primarily to using hello messages to maintain OSPF neighbor relations. Turning link metrics on even with low hello and dead intervals (1/4 ON) increased control packets and dropped reach slightly due to conflicting neighbor up/down signaling provided by the radio and the hello/dead interval kicking in.

TABLE IX
TRADEOFF COMPARISON (ALL TO ALL NODES - DAY 2

	1/4 OFF	1/4 ON	10/40 OFF	10/40 ON
Ctrl Pkts (Hello)	641.8 KB	640.6 KB	90.5 KB	95.4 KB
Ctrl Pkts (LSA)	535.8 KB	682.2 KB	468.3 KB	573.8 KB
Ctrl Pkts (DD)	53.2 KB	49.2 KB	24.4 KB	32.3 KB
Total Ctrl Pkts	1231 KB	1372 KB	593 KB	701.5 KB
Reach (Routes)	85.3%	81.4%	78.9%	90.2%
Reach (Ping all)	69.6%	62.2%	50.3%	58.9%
RTT (Ping all)	162.18ms	134.23ms	134.26ms	120.17ms

By lowering the periodicity of the hello and dead interval to 10/40 and turning link metrics off (10/40 OFF), the overhead from control packets was significantly reduced, but reachability as described by kernel routes as well as actual pings across the network dropped significantly. The final set of statistics (10/40 ON) was the default configuration we ran throughout the whole exercise and our primary demonstration case. It consists of using link metrics to do the primary route cost calculation and the neighbor up/down signaling, while relying on OSPF hellos only as a last resort, sending hellos only once ever 10 seconds. Unfortunately, the results show that although the route availability increased, there was a 10% drop in ping reachability as compared to simply turning off OSPF cross-layer information and pushing the hello and dead intervals to 1/4. Overall, it was difficult to conclusively demonstrate that using pure link metrics to determine reachability performed better than each of the other cases under conditions of varying aircraft positions, weather conditions, and software issues. It is recommended to re-try these tests in a more controlled environment such as lab emulation.

E. Link vs. Path Availability Analysis

While link uptime and statistics are helpful in understanding point-to-point performance, end-to-end path statistics evaluate the overall impact on the network from the routing protocols. In order to fully understand the viability of sending application data through a network, end-to-end characterization is a necessity. In this section, we evaluate the routing performance by examining end-to-end path data including the end-to-end network RTT and uptime, preferred path taken, and number

of path changes per hour. Specifically, we examine the performance on both flight days.

1) End-to-End Network Uptime Results: A key indicator in network performance is end-to-end uptime. Today's internet applications were primarily designed to operate over low packet loss networks. In cases of high loss and sporadic outages, these applications often fail. In this section, we examine the end-to-end network uptime and average latency. Understanding trends in an airborne network provides an important baseline for which to design applications that function in highly disruptive environments. To measure the network uptime, timestamped 16Byte IPv4 pings were sent from logger machines on each platform to each of platforms in the network and the results were logged over the onstation time of each day. The forward and reverse path of the ping can potentially be different as OSPF cost mismatch is possible using link metrics. IPv4 pings over Iridium and the landlines were systematically dropped. Average OSPF cost was calculated by averaging over the path cost as reported by the Linux kernel routing table. For OSPF cost and ping RTT, 95% confidence intervals of the values were reported as well.

TABLE X
COMP. OF PATH COST, RTT, AND NETWORK UPTIME - DAY 1

Platform Src/Sink	Avg OSPF Cost	Avg Ping RTT	Uptime
Trailer - SAIL	72.0 ± 5.09	84.9 ± 7.95ms	91.46%
SAIL - Trailer	70.0 ± 5.25	85.7 ± 8.30ms	90.49%
RC12L - Trailer	44 ± 0.45	138.6 ± 1.79ms	67.64%
Trailer - RC12L	48 ± 1.36	134.9 ± 1.78ms	67.27%
RC12L - SAIL	112 ± 5.76	230.3 ± 10.6ms	62.3%
SAIL - RC12L	113 ± 6.0	234.6 ± 11.2ms	62.1%

Tables X and XI show the average OSPF path cost and end-to-end ping RTT and network uptime for Day 1 and Day 2. It can be seen that from the Trailer to the SAIL, very high network uptime (91.46%) was achieved indicating that routing over the HCB was highly successful. The RC12L had only solid connections from itself to the Trailer with roughly 67-69% availability. This resulted in an RC12L to SAIL end-to-end connectivity of a combination of 86-91% availability from Trailer to SAIL and 67-69% availability from RC12L to Trailer which was slightly less than both at between 55-62%. End-to-end RTT on average was in the hundreds of milliseconds with 95% confidence intervals less than tens of milliseconds while it was indicated that DRS latency averaged 2ms. The results indicate utilization of multiple links with varying latencies and link availability.

TABLE XI
COMP. OF PATH COST, RTT, AND NETWORK UPTIME - DAY 2

Platform Src/Sink	Avg OSPF Cost	Avg Ping RTT	Uptime
Trailer - SAIL	87.0 ± 10.0	103 ± 15.2ms	86.9%
SAIL - Trailer	99.0 ± 10.8	98.5 ± 15.2ms	87.1%
RC12L - Trailer	17 ± 1.84	145.3 ± 10.18ms	69.64%
Trailer - RC12L	21 ± 1.83	131.1 ± 7.61ms	69.98%
RC12L - SAIL	118 ± 12.0	273 ± 24.3ms	55.3%
SAIL - RC12L	128 ± 12.3	244 ± 21.2ms	56.7%

2) End-to-End Path Choice Results: In addition to network uptime and RTT, it is interesting to characterize the rate of path changes in a network as well as the amount of time spent in each permutation. We define a permutation as a set of links taken from one platform to another. An example of a permutation between the Trailer and SAIL is going through DRS from the Trailer to the PR and DRS from the PR to the SAIL (DRS - DRS). In this subsection, we examine the total hop and permutation change rate as well as the preferred path permutation choice between the Trailer and SAIL and RC12L and SAIL on Day 1 and Day 2. We limited ourselves to the Trailer to SAIL and RC12L to SAIL because the Trailer and SAIL represent the two ground entry points of the network while the RC12L was the gateway to all the tactical data information provided by the BQ09 participants. Figures 16 and 18 show the path preference between the Trailer and SAIL on Day 1 and Day 2 while Figures 17 and 19 show the path preference between RC12L and SAIL.

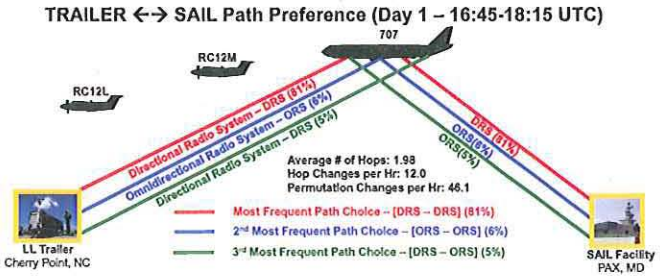


Fig. 16. Trailer to SAIL path preference on Day 1. The most frequent path choice was DRS - DRS at 81% of uptime.

As can be seen in Figure 16, the primary path taken between the Trailer and the SAIL facility was DRS from the Trailer to PR, and DRS from PR to SAIL. Although we had hoped that an end-to-end link between the Trailer and the SAIL facility would be established through ESB air-to-air, the uptime was significantly lower. The average number of hops was 1.98 suggesting at times the trailer preferring the landline between SAIL and the TR as its primary path when no HCB paths were available.

Figure 16 also shows that the number of hops changes were about 12.0 per hour (approximately once every 5 minutes) and the number of permutation changes to be 46.1 per hour (approximately once every 1.3 minutes). Since some applications require atleast 20-25 seconds of uptime to be able to successfully operate, changing permutations once every 1.3 minutes meets this minimum requirement.

Following the previous analysis between the Trailer and the SAIL, Figure 19 gives the same statistics between RC12L and the SAIL. On Day 1, the primary link between the RC12L and any other platform was ESB to TR and as such, its path choice was fairly similar to those shown in Figure 18 with the addition of an R2 between RC12L and the Trailer. Its interesting to note that RC12M played very little if any role in passing actual traffic. This was perhaps due to the stability of the ORS and DRS links and the preference given them due to reported PADQ data rate and latency.

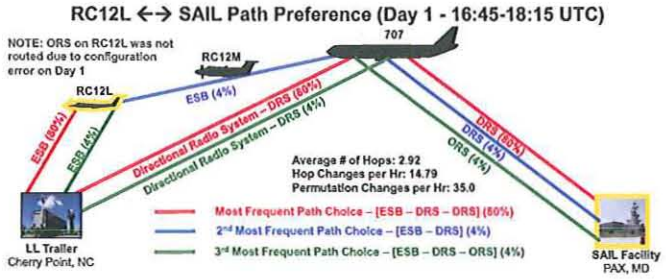


Fig. 17. RC12L to SAIL path preference on Day 1. The most frequent path choice was ESB - DRS - DRS at 80% of uptime.

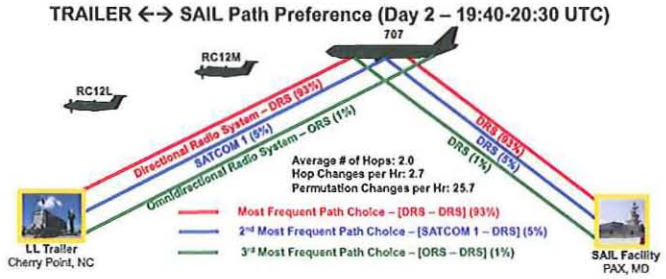


Fig. 18. Trailer to SAIL path preference on Day 2. The most frequent path choice was DRS - DRS at 93% of uptime.

Figure 18 illustrates the path preference between the SAIL and Trailer on Day 2. The permutation choices remained fairly similar to the previous day, but the average permutation changes seem to have dropped significantly (25.7 permutation changes/hr on Day 2 vs. 46.1 permutation changes/hr on Day 1). This was due to a modification of OSPF weighting. On Day 1, the OSPF calculation weighted latency and CDR to 50/50 while on Day 2, it was modified to favor CDR at 75/25. This was done to prevent link flapping between ESB and ORS, favoring ESB over ORS. The result was less OSPF SPF calculations and path changes.

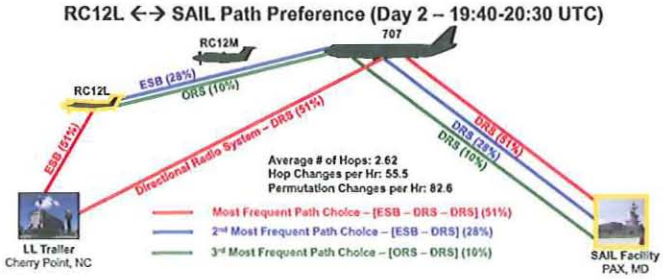


Fig. 19. RC12L to SAIL path preference on Day 2. The most frequent path choice was RC12L ESB to TR, TR DRS to PR, and PR DRS to SAIL at 51%.

Figure 19 shows the end-to-end path choices taken from RC12L to the SAIL. As can be seen, RC12L favored going ESB to TR and then DRS - DRS from TR to SAIL at 51%. The next preferred link was ESB to PR and then DRS down to SAIL. There was a small percentage of the time when RC12L chose ORS to PR and then DRS down to SAIL. This was due to the fact that ESB had periodic body blockage while ORS provided almost constant availability. Due to this switch between ESB and ORS as was not the case with Day 1 due to a misconfiguration leading to ORS not being used on RC12L, OSPF path changes were more frequent. The average hop changes per hour were at 55.5 (almost once a second) while the permutation changes spiked at an average of 82.6 per hour (almost once ever 43 seconds). The high number of routing changes no doubt led to difficulty of data passing end-to-end from RC12L to SAIL.

IV. Conclusion

The goal of designing, building and demonstrating a stable airborne layer high capacity network backbone is still very much a work in progress. In this paper, we evaluate several of the radio technologies available in an airborne environment and demonstrate the viability of the RFC4938 radio-to-router interface against open source platforms. This work feeds back into simulation and emulation work in better determining the technical and operational requirements of an aerial network as well as provides tools and open source implementations of platforms that can be tested against by others. Key technical measurement successes include:

- Successful demonstration of the common radio-to-router interface functionality in a heterogeneous airborne environment.
- Successful demonstration of the open source router functionality modified to support MDR, address families, and other extensions
- Successful demonstration of end-to-end connectivity through an aerial layer of 250 Nm over 3 aerial platforms with heterogeneous radios.

Although the flight test provided an interesting performance evaluation framework for various radio systems, there are several avenues of future work to pursue including stabilization of the software issues we encountered during the test, development of a broadcast radio standard to complement RFC4938, and additional measurements with other high capacity radio systems.

References

[1] B. Berry and H. Holgate, "Ppp over ethernet (pppoe) extensions for credit flow and link metrics," Internet Engineering Task Force, RFC 4938, 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4938.txt

[2] "Mobile ad hoc networks for router-to-radio communications," Cisco Systems, Tech. Rep., 2007. [Online]. Available: http://www.cisco.com/en/US/docs/ios/12_4t/ip_mobility/configuration/guide/ip_manet.html

[3] "Quagga open source router source code," GPLv2, Tech. Rep. [Online]. Available: http://www.quagga.net

[4] R. Ogier and P. Spagnolo, "obile ad hoc network (manet) extension of ospf using connected dominating set (cds) flooding," Internet Engineering Task Force, RFC 5614, 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5614.txt

[5] A. R. M. B. A. Lindem, S. Mirtorabi and R. Aggarwal, "Support of address families in ospfv3," Internet Engineering Task Force, RFC Draft 10, 2009. [Online]. Available: http://tools.ietf.org/html/draft-ietf-ospf-af-alt-10

[6] J. Macker, "Simplified multicast forwarding," Internet Engineering Task Force, RFC Draft 9, 2009. [Online]. Available: http://tools.ietf.org/html/draft-ietf-manet-smf-09